REMARKS

The following remarks are deemed fully responsive to the outstanding office action mailed April 7, 2006. Claims 1 – 10, 12, 13, 15, 16 and 18 are pending, of which claims 1, 12, 15 and 18 are independent.

Claim Rejections – 35 U.S.C. § 102

Claims 1 – 10, 12, 13, 15, 16 and 18 stand rejected under 35 U.S.C. § 102(e) as being taught by Intel's publication "Hyper-Threading Technology" in Intel Technology Journal: Volume 06 Issue 01, published February 14, 2002 (hereinafter "Intel"). Applicant respectfully disagrees.

As Applicant noted in his response of November 9, 2005, the immediate application detects and then distributes bundled instructions to clusters according to threads associated with the instructions. On the other hand, Intel does not disclose instruction bundling at all. The architecture of Intel, as shown in Figure 3, Processors with Hyper-Threading Technology, page 7, col. 1, is patentably distinct from that of the immediate application, as shown in FIG. 3 of the drawings. In particular, processor 106 of the immediate application has multiple clusters 102, each with multiple pipelines 103, associated with a program counter 104 and a thread decode unit 130. On the other hand, as disclosed on page 7, Hyper-Threading Technology Architecture, col. 1, paragraph 3, the processor of Intel duplicates general-purpose registers, control registers, an advanced programmable interrupt controller and some machine state registers, but does not include multiple clusters of execution units.

To anticipate a claim, Intel must teach every element of the claim and "the identical invention must be shown in as complete detail as contained in the ... claim." *MPEP 2131* citing *Verdegaal Bros. V. Union Oil Co. of California*, 814 F.2d 628, 2 USPQ2d 1051 (Fed. Cir. 1987) and *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 9 USPQ2d 1913 (Fed. Cir. 1989). Intel does not teach every element of claims 1 – 10, 12, 13, 15, 16 and 18.

Claim 1 recites a method for processing bundled instructions through execution units of a processor, including the steps of:

a) determining a throughput mode of operation, based upon a configuration bit;

b) fetching a first bundle of singly-threaded instructions from a singly- or multiply-threaded program;

c) distributing the first bundle to a first cluster of the execution units for execution therethrough;

d) fetching a second bundle of singly-threaded instructions from the program; and

e) distributing the second bundle to a second cluster of the execution units for execution therethrough.

In step a) of claim 1, a configuration bit is used to determine a throughput mode of operation. The Examiner asserts that Intel, on page 12, in the section titled 'Single-Task and Multi-Task,' discloses features of step a). Respectfully, we disagree. Intel instead teaches away from step a), disclosing that "on a processor with Hyper-Threaded technology, executing HALT transitions the processor from MT-mode to ST0- or ST1-mode, depending on which logical processor executed the HALT." See Intel page 12, col. 2 second paragraph. Again teaching away from step a), Intel discloses that "in ST0- or ST1-modes, an interrupt sent to a HALTed processor would cause a transition to MT-mode." See Intel, page 12, column 2, third paragraph. The mode transition of Intel is not based upon a configuration bit. In fact, Intel does not disclose a configuration bit with relation to single-task and multi-task modes of operation.

In one example of the immediate application, a bundle contains three instructions as described in paragraph [0001]. In a superscalar processor it is not uncommon to execute a large window of instructions out-of-order, as described in Intel on page 5 lines 9-13. But, Intel does not disclose or suggest bundling instructions, nor fetching singly-threaded instructions as required by step b). The window of Intel is not equivalent to bundling instructions. Further, Intel does not disclose or suggest distributing bundled instructions to a specific cluster, as required by step c). Intel also does not disclose or suggest fetching a second bundle of singly-threaded instructions, as required by step d), and does not disclose or suggest distributing this second bundle of instructions to a specific cluster, as required by step e). Intel does not disclose bundling of instructions.

As required by steps c), d) and e) of claim 1, bundles of singly- or multiply-threaded instructions are distributed to two clusters, thereby increasing instruction processing performance. On the other hand, Intel discloses that if only one thread is being processed, Hyper-Threaded Technology does not have benefit and must recombine partitioned system resources in order to ensure no reduction in speed performance. See Intel page 7, First Implementation on the Intel Xeon Processor Family, col. 2, paragraph 3. Intel does not disclose processing singly threaded instructions through multiple clusters.

For at least these reasons, Intel cannot anticipate claim 1. Reconsideration of claim 1 is respectfully requested.

Claims 2-10 depend from claim 1 and benefit from like argument. However, these claims have additional features that patentably distinguish over Intel. For example, claim 2 recites processing the first bundle within the first cluster. Claim 3 recites processing the second bundle within the second cluster. As argued above, Intel does not disclose processing bundled instructions within clusters. Further, again as argued above, Intel does not utilize multiple clusters for singly threaded programs.

Claim 4 recites architecting data from the first cluster to a first register file. Claim 5 recites committing architected state from the second cluster to the first register file. Claim 6 recites architecting data from the second cluster to a second register file. As argued above, Intel does not disclose using clusters. Further, Intel does not disclose first and second register files that correspond to each cluster, as described in paragraph [0013] of the specification and shown in FIG. 3 of the drawings. Teaching away from claims 4, 5 and 6, on page 7, Hyper-Threading Technology Architecture, col. 1, paragraph 3, Intel discloses that a complete set of the architectural state, including register files, for each of two logical processors is maintained; but Intel does not disclose architecting data from the second cluster to the first register file, as required by claim 5. In fact, Intel does not disclose clusters of pipelines within a single processor anywhere.

Claim 7 recites decoding instructions into the first bundle of the singly-threaded instructions. Claim 8 recites decoding instructions into the second bundle of the singly-threaded instructions. Intel does not disclose decoding singly-threaded

instruction into a first or second bundle, and therefore cannot teach limitations of claims 7 and 8.

Claim 9 recites fetching a third bundle of singly-threaded instructions, distributing the third bundle to the first and second clusters of the execution units for execution therethrough, and bypassing data between the clusters, as needed, to facilitate processing of the third bundle through the clusters. Claim 10 recites utilizing a latch to couple the data between the clusters. Intel does not teach fetching, distributing or bypassing data between clusters for bundled instructions, and therefore cannot teach limitations of claims 9 and 10.

In view of the above arguments, Intel cannot anticipate claims 2-10. Reconsideration of claims 2-10 is respectfully requested.

Claim 12 recites a method for processing bundled instructions through execution units of a processor, including the steps of:

a) fetching a first bundle of singly-threaded instructions;
b) distributing the first bundle to two or more clusters of the execution units for execution therethrough; and
c) bypassing data between the clusters, as needed, to facilitate processing of the first bundle through the clusters.

The logical processor of Intel does not have multiple clusters of execution units. As argued above, each logical processor of Intel has only architecture states and not execution units duplicated. Again, Intel does not disclose or suggest fetching a first bundle of singly-threaded instructions, as required by step a). Further, Intel cannot disclose or suggest distributing bundled instructions to two or more clusters, as required by step b), since Intel does not disclose clusters of instruction processing pipelines. Therefore, Intel cannot disclose bypassing data between clusters as required by step c).

For at least these reason, Intel cannot anticipate claim 12. Reconsideration of claim 12 is respectfully requested.

Claim 13 depends from claim 12 and benefit from like argument. However, this claim has additional features that patentably distinguish over Intel. For example, claim 13 recites fetching a second bundle of singly-threaded instructions, distributing

the second bundle to one of the clusters for execution therethrough, fetching a third bundle of singly-threaded instructions, and distributing the third bundle to another one of the clusters units for execution therethrough. Again, Intel does not disclose or suggest fetching a bundle of singly-threaded instructions or of distributing bundles of instructions to clusters of execution units.

Intel cannot, therefore, anticipate claim 13. Reconsideration is requested.

Claim 15 recites a processor architecture of the type having two or more clusters of execution units for processing instructions, the improvement including a thread decoder for grouping instructions into singly threaded bundles and for distributing the bundles to the clusters according to either a wide mode or throughput mode of operation. As argued above, Intel does not disclose or suggest multiple clusters of execution units or of processing bundled instructions within these clusters. Intel further does not disclose a thread decoder for grouping instructions into singly threaded bundles. Further again, Intel does not disclose or suggest multiple modes of operation for singly threaded bundles.

For at least these reasons, Intel cannot anticipate claim 15; thus, reconsideration is respectfully requested.

Claims 16 and 17 depend from claim 15 and benefit from like argument. However, these claims have additional features that patentably distinguish over Intel. For example, claim 16 recites each cluster comprises a core and register file. Intel does not disclose clusters. Claim 17 recites the thread decoder distributes bundles of singly-threaded instructions through multiple clusters in the wide mode of operation, and the thread decoder distributes bundles of singly-threaded instructions through one of the clusters in the throughput mode of operation. Intel does not disclose clusters. Further, Intel does not disclose a thread decoder. Intel also does not disclose or suggest two modes of operations wherein singly-threaded instructions are distributed to one or more clusters.

For at least these reasons Intel cannot anticipate claims 16 and 17. Reconsideration of claims 16 and 17 is respectfully requested.

Claim 18 recites a method for processing bundled instructions through execution units of a processor, including the steps of:

a) determining, based upon a configuration bit, a throughput mode or wide mode of operation;

b) fetching a first bundle of singly-threaded instructions from a singly- or multiply-threaded program;

c) if in throughput mode of operation, distributing the first bundle to a first cluster of the execution units for execution therethrough;

d) if in wide mode of operation, distributing the first bundle to multiple clusters of the execution units for execution therethrough;

e) fetching a second bundle of singly-threaded instructions from the program;

f) if in throughput mode of operation, distributing the second bundle to a second cluster of the execution units for execution therethrough; and

g) if in wide mode of operation, distributing the second bundle to multiple clusters of the execution units for execution therethrough.

As argued above, Intel does not disclose or suggest determining a throughput mode of operation based upon a configuration bit as required by step a) of claim 18. Further, Intel does not disclose fetching a first bundle of singly-threaded instructions as required by step b) and does not disclose distributing that first bundle of instructions based on throughput or wide mode of operation, as required by steps c) and d). Further again, Intel does not teach fetching a second bundle of singly-threaded instructions as required by step e) and does not disclose distributing that second bundle of instructions based on throughput or wide mode of operation, as required by steps f) and g). Intel does not disclose clusters.

For at least these reasons, Intel cannot anticipate claim 18. Reconsideration of claim 18 is respectfully requested.

Applicant believes no fees are due in connection with this Response; however, if any fee is deemed necessary, the Commissioner is authorized to charge such fee to Deposit Account No. 08-2025.

Respectfully submitted,
LATHROP & GAGE, LC.

By: 

Dan Cleveland, Jr. , Reg. No. 36,106
4845 Pearl East Circle, Suite 300
Boulder, Colorado 80301
Telephone: (720) 931-3012
Facsimile: (720) 931-3001